

Vue Lecture-2 By Louis Chuang

Vue Lecture-2

2020/03/04 (三)

Installation



❖ 安裝vue-cli

❖ 1. 安裝NodeJS »



❖ 2. 安裝vue-cli

❖ Mac » `sudo npm install -g vue-cli`

❖ Win » `npm install -g vue-cli`

❖ 通用平台考量，講座中使用VSCODE

❖ 1. 安裝VSCODE »

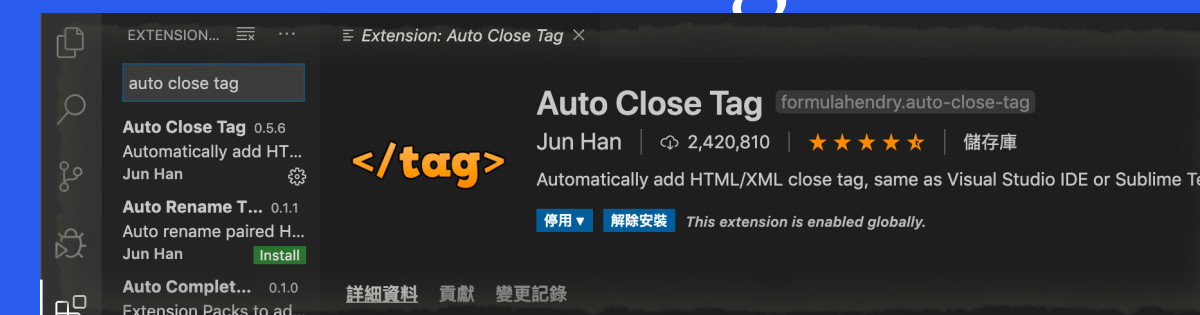


❖ 2. 安裝Plugin For VSCODE

❖ a. Vetur



❖ b. Auto Close Tag



Lecture Resources

- ❖ Lecture2Resource([點此下載](#))
- ❖ 演示範例: Resource-Demo
- ❖ 練習資源: Resource-Workout



Goals

- ❖ 建構VueJS基礎知識，減少自主學習成本
- ❖ 講座過程即時實作建立概念
- ❖ 建立一個Single Page Application



Agenda

- ❖ 回顧Lecture-1
- ❖ Forms
- ❖ Filters & Mixins
- ❖ Single Page Application
 - ❖ Routing
- ❖ State Management
 - ❖ Vuex



Review

回顧上週內容

- ❖ Vue Instance
 - ❖ data
 - ❖ methods
 - ❖ computed
 - ❖ watched
 - ❖ lifecycle
 - ❖ multiple vue instance

Review

回顧上週內容

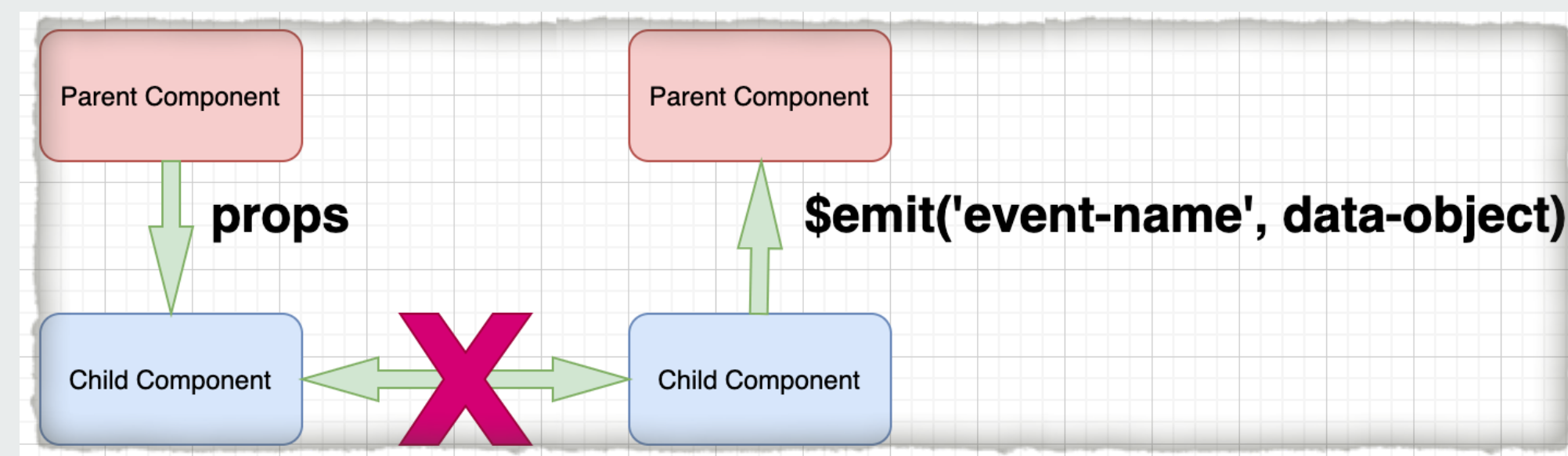
- ❖ Template Syntax
 - ❖ v-bind
 - ❖ v-on
 - ❖ modifier (submit.prevent)
 - ❖ v-model
- ❖ Single-file Component
 - ❖ .vue的文件
 - ❖ 避免字串模板

Review

回顧上週內容

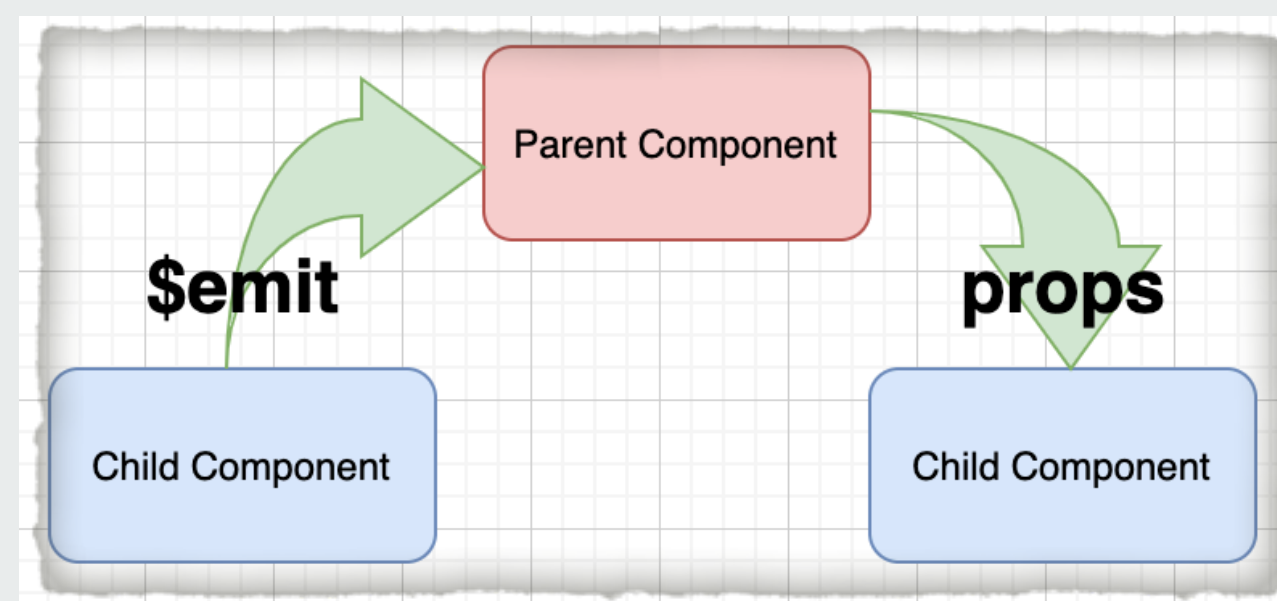
❖ Component Communication

Parent \longleftrightarrow Child Communication

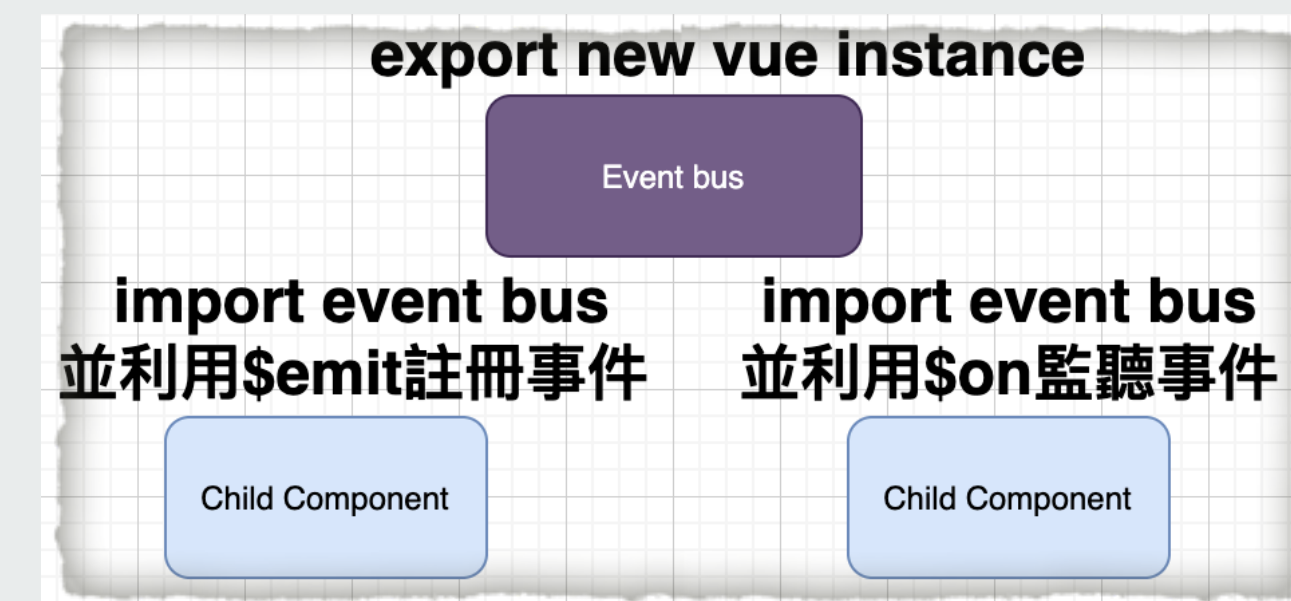


Child \longleftrightarrow Child Communication

(A)



(B)



Review

回顧上週內容

- ❖ Development Workflow
 - ❖ 安裝步驟
 - ❖ NodeJS
 - ❖ Vue CLI
 - ❖ 使用Vue CLI建立Webpack-Simple Project
 - ❖ 專案結構介紹
 - ❖ 開始使用Single-File-Template

Forms – User Input

vue-lecture2 / demo1-forms

V-MODEL

- ❖ 利用v-model與User Input元件進行雙向綁定
 - ❖ TEXT
 - ❖ TEXT AREA
 - ❖ CHECKBOX
 - ❖ RADIO BUTTON
 - ❖ DROPDOWN

Vue Lecture-2

動動手» Workout-1

開啟workout1-forms進行實作



Filters & Mixins

Filters

vue-lecture2 / demo2-filters

- ❖ 在Template當中使用，將資訊進行客製化轉換並呈現
- ❖ 實際上只會轉換呈現資訊而不會改變data原始的資料
- ❖ 使用Filters過濾List資訊
 - ❖ 在Vue1.X版本有提供一些Filter可以進行分頁過濾等等，但在Vue2.X版本已被移除
 - ❖ 可使用Computed實作

Vue Lecture-2

動動手» Workout-2

開啟workout2-filters進行實作



Filters & Mixins

vue-lecture2 / demo3-mixins

Mixins

- ❖ 建立可重複使用的功能
 - ❖ 可包含任意物件，data, methods, components...
- ❖ 不會銷毀任何東西（資料狀態不會遺失）
- ❖ 透過Merges的方式將Mixin的功能加入到已存在的Vue Instance
- ❖ 使用Mixin，載入有順序性，Mixin > Components
- ❖ 定義在Mixin當中的物件資料不會共享，而是複製在各個Component

Vue Lecture-2

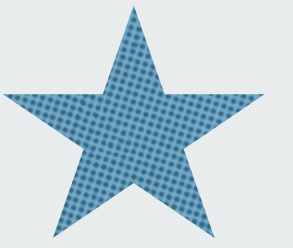
動動手» Workout-3

開啟workout3-mixins進行實作





Single Page Application - Routing

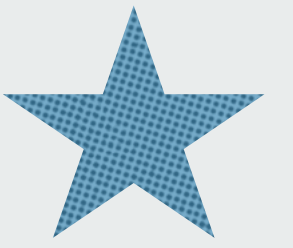


什麼是Single Page Application ?

- ❖ 通過動態重新複寫目前頁面與使用者的操作進行互動
- ❖ VueJS透過vue-router來達到這個目的



Single Page Application - Routing



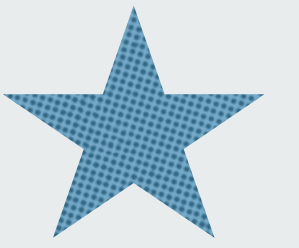
vue-lecture2 / demo4-router

配置Vue-Router & Url Hash 模式

- ❖ 安裝vue-router
 - ❖ npm install vue-router —save
- ❖ 配置
 - ❖ Vue.use(VueRouter)來新增Plugin
 - ❖ 宣告並註冊Router在Vue Instance
 - ❖ 需要被載入的component透過router-view指定render的位置
- ❖ Hash模式 » http://host:port/#/page
 - ❖ 確保page請求能被Javascrip Application攔截處理，使之不會重新加載
 - ❖ 使用html 5 history模式利用history.pushState API達到URL改變不會重新加載頁面



Single Page Application - Routing



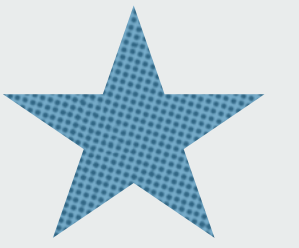
vue-lecture2 / demo4-router

定義Routes配置指定給Router Instance

- ❖ 定義格式 » { **path**: '<Your Route>', **component**: <Your Component> }
- ❖ 設定route目的 » 匹配該路由需要取得的資源(Component)，並透過<router-view></router-view>渲染呈現
- ❖ 屬性
 - ❖ path » 定義路由
 - ❖ component » 匹配的component
- ❖ Nested Route » 嵌套路由
 - ❖ 使用children: [{path.....}] 定義
 - ❖ Nested Route匹配指定的component後，一樣透過<router-view></router-view>渲染呈現頁面



Single Page Application - Routing



Router-Link

vue-lecture2 / demo5-router-link

- ❖ 預設渲染產生<a>標籤並自帶routing成功後的style
- ❖ 屬性
 - ❖ to » 指定目標地址 （將值傳至\$router.push()，可以傳入字串或是物件 ex: { path: '/', params: { id: 1 } })
 - ❖ tag » 產生指定標籤
 - ❖ active-class » 自定義routing成功後的Style
 - ❖ exact » 表示精準匹配位址才會Active Style
 - ❖ 更多詳細請查閱文檔
- ❖ 取得Route parameter: 定義vue-router時，可使用props傳遞參數 (點擊查閱範例)
- ❖ 優化Router-Link的方式 » router定義中使用name關鍵字為Router命名，更容易維護 ({ name: 'RouterName' })
- ❖ 如何進行Redirect» router定義中使用redirect關鍵字

Vue Lecture-2

動動手» Workout-4

開啟workout4-router進行實作





State Management - Vuex



在大型專案中使用Vuex管理狀態 - Centralized State

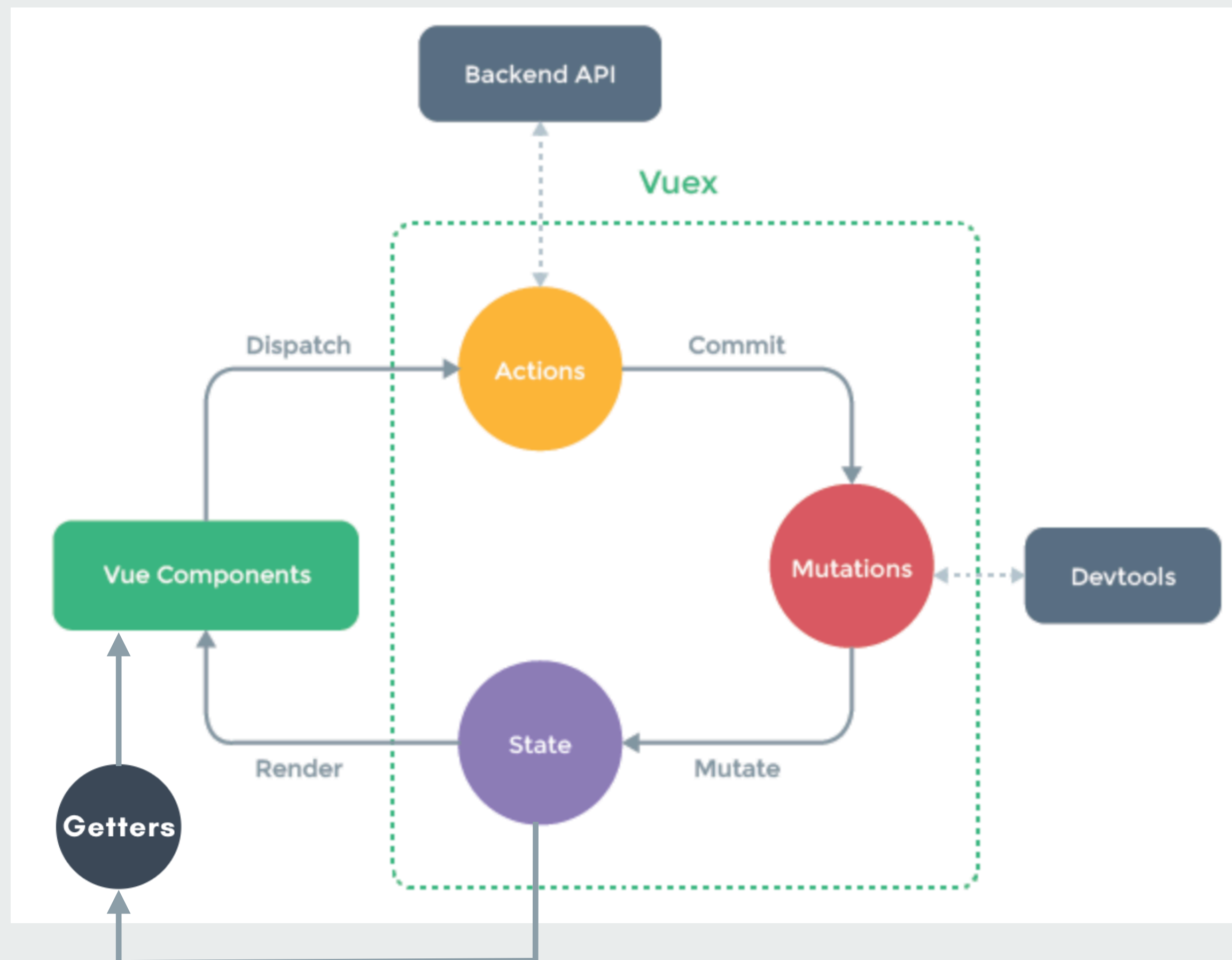
- ❖ 大型專案中有許多Component，當Component之間資訊需要進行共享時有以下問題
 - ❖ 有太多狀態需要\$emit
 - ❖ 資料異動不容易追蹤
- ❖ Vuex使用單一樹狀結構提供我們以下優勢
 - ❖ 中心化管理狀態（一個應用只會有一個Store Instance）
 - ❖ 容易管理並追蹤狀態



State Management - Vuex



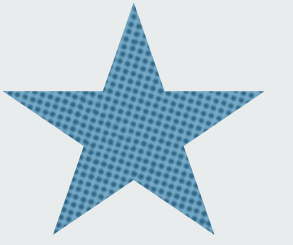
Vuex運作流程



- ❖ State » Central Store
- ❖ Getters » Get State From Central Store
 - ❖ mapGetters (from Vuex)
- ❖ Mutations » **Synchronous Commit**
 - ❖ mapMutations (from Vuex)
- ❖ Actions » **Dispatch Asynchronous Commit**
 - ❖ mapActions (from Vuex)



State Management - Vuex

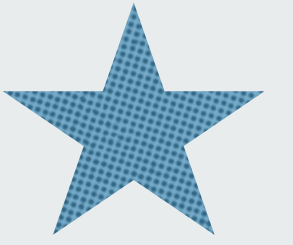


配置並使用Vuex

- ❖ 安裝vuex
 - ❖ `npm install vuex —save`
- ❖ 配置
 - ❖ `Vue.use(Vuex)`來新增Plugin
 - ❖ 宣告並註冊Store在Vue Instance



State Management - Vuex



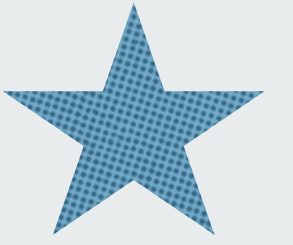
取得狀態state - Getters

vue-lecture2 / demo6-vuex-getters

- ❖ Vuex 在 Vue Instance提供\$store讓我們能夠控制Central Store
- ❖ 取得狀態 » `this.$store.state.<your data variable>`
- ❖ 透過定義getters來取得狀態
 - ❖ 定義getter function於Central Store的getters中
 - ❖ 任意地方取得狀態 » `this.$store.getters.<Your getters functions name>`
- ❖ 使用mapGetters直接引入Central Store的getters



State Management - Vuex



vue-lecture2 / demo7-vuex-mutations

更新狀態 Mutations & Actions

- ❖ Mutations 必須是 **Synchronous Commit**
- ❖ 透過定義 mutations 來更新狀態
 - ❖ 定義 mutations function 於 Central Store 的 mutations 中
 - ❖ 任意地方利用 store.commit 更新狀態 » `this.$store.commit('<Your mutations functions name>')`
- ❖ 使用 mapMutations 直接引入 Central Store 的 mutations



State Management - Vuex



vue-lecture2 / demo8-vuex-actions

更新狀態 Mutations & Actions

- ❖ Actions 必須是 **Dispatch Asynchronous Commit**
- ❖ 透過定義actions來更新狀態
 - ❖ 定義actions function於Central Store的actions中
 - ❖ 任意地方利用store.dispatch觸發，並且透過context.commit至mutations進行更新狀態
 - ❖ this.\$store.dispatch('<Your actions functions name>')
 - ❖ 在actions function當中 » context.commit('<Your mutations functions name>')
- ❖ 使用mapActions直接引入Central Store的actions



State Management - Vuex



vue-lecture2 / demo9-vuex-vmodel

Special Case

- ❖ Vuex中使用Two-Way-Binding (v-model)需特別處理
 - ❖ computed properties 當中自定義get() & set()

Vue Lecture-2

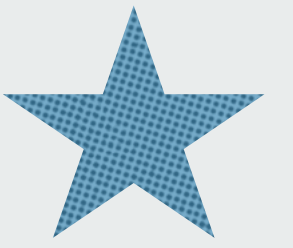
動動手» Workout-5

開啟workout5-vuex進行實作





Bonus - Deploy Vue Application



部署流程介紹

- ❖ VueCLI建置專案已經提供了部署所需配置
- ❖ 講座中使用到的是Webpack3
 - ❖ 於webpack.config.js當中配置DefinePlugin

```
module.exports.plugins = (module.exports.plugins || []).concat([
  new webpack.DefinePlugin({
    'process.env': {
      NODE_ENV: '"production"'
    }
  })
],
```

- ❖ 當運行Production Mode時會移除掉在開發時期才需要的警告提示，可以讓檔案變得更精簡
- ❖ 編譯後src folder的檔案會統一打包在在dist/build.js
- ❖ index.html檔案也需要部署

「謝謝百忙之中還到場參與的各位。」

–*Louis Chuang*